

09/29/00  
JC915 U.S. PTO

10-2-00

A

ASSISTANT COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, DC 20231

PATENT  
Date: September 29, 2000  
File No. 3408.64755

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): Katsumi Shirai, Kiyotaka Fukawa

For: EMULATION PROCESSING METHOD  
FOR A STORAGE DEVICE AND STORAGE  
DEVICE

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on this date.

9-29-00  
Date

Express Mail Label No. EL409506435US

Enclosed are:

- (X) 26 pages of specification, including 18 claims and an abstract.
- (X) an executed oath or declaration, with power of attorney.
- ( ) an unexecuted oath or declaration, with power of attorney.
- ( ) sheet(s) of informal drawing(s).
- (X) 9 sheet(s) of formal drawings(s).
- (X) Assignment(s) of the invention to Fujitsu Limited.
- (X) Assignment Form Cover Sheet.
- (X) A check in the amount of \$40.00 to cover the fee for recording the assignment(s) is enclosed.
- (X) Information Disclosure Statement.
- (X) Form PTO-1449 and cited references.
- ( ) Associate power of attorney.
- (X) Priority Document.

JC930 U.S. PTO  
09/29/00  
09/29/00


Fee Calculation For Claims As Filed

a) Basic Fee						\$ 690.00
b) Independent Claims	2	-	3	=	0	x \$ 78.00 = \$
c) Total Claims	18	-	20	=	0	x \$ 18.00 = \$
d) Fee for Multiple Claims						\$260.00 = \$
Total Filing Fee						\$ 690.00

- (X) A check in the amount of \$ 690.00 to cover the filing fee is enclosed.
- (X) The Commissioner is hereby authorized to charge any additional fees which may be required to this application under 37 C.F.R. §1.16-1.17, or credit any overpayment, to Deposit Account No. 07-2069. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 07-2069. A duplicate copy of this sheet is enclosed.

Suite 8660 - Sears Tower  
233 S. Wacker Drive  
Chicago, Illinois 60606  
(312) 993-0080

GREER, BURNS & CRAIN, LTD.

By:   
Patrick G. Burns  
Registration No. 29,367

9-29-00  
Date

Express Mail Label No.: EL409506435US

## TITLE OF THE INVENTION

Emulation Processing Method for a Storage Device and Storage Device

## BACKGROUND OF THE INVENTION

### Field of the Invention

This invention relates to a storage device and an emulation processing method for accessing a storage medium having a first sector length unit by a command with a second sector length unit that is shorter than the first sector length unit, and more particularly to a storage device and an emulation processing method for the storage device for protecting data of an address other than the target address from a write error.

### Description of the Related Art

As host performance improves, the capacity of storage media, such as disks, is doubling every year. Due to this increase in capacity of the storage medium, the sector length has become longer. However, there are some host OS that do not correspond to the change in sector length. Therefore, emulation technology, that can size-convert to the sector length units of the storage medium by a command with a sector length of the host OS, is necessary.

For a disk such as an optical disk, reading and writing is performed in sector units. This sector is broken up by a sector ID, and is set a user area (data area) having a specified length. For example, a sector length of 512 bytes is mainstream. However, with the recent demand for larger capacity disks, disks with a

sector length of 1024 bytes or 2048 bytes are now provided. Increasing the sector length in this way makes it possible to increase the user area, and thus increase the disk capacity.

On the other hand, the OS (file system) that uses this disk defines the sector size and specifies the address. For example, in the UNIX system, the recognized sector size is 512 bytes. In a file system such as UNIX, a 512-byte sector size disk was used. However, it is desired to be able to use large-capacity disks (2048 bytes) as mentioned above with a file system such as UNIX as well.

In order to use a 2048-byte sector disk in this environment with a 512-byte sector, so-called emulation technology that converts the sector size has been proposed (for example, Japanese Unexamined published patent No. H4-165527, USP No. 5485439, Japanese Unexamined published patent No. H4-14127 and Japanese Unexamined published patent No. H9-146710). Fig. 10 and Fig. 11 explain this prior emulation process.

Fig. 10 shows a file system that uses a magneto-optical disk device 91 in which the disk medium 92 stores data in sector length units of 2048 bytes (first sector length). On the other hand, the OS of the host 90 accesses a file in 512-byte units (second sector length). The relationship between the sectors recognized by this host and the actual sector size of the disk is as shown in Fig. 11. In other words, the sector addresses of the host for the actual sectors of sector address 'a' of the disk are: 4a, 4a+1, 4a+2 and 4a+3.

Conversion of this sector size is performed as follows. First, the sector address from the host 90 is converted to the actual

sector size of the disk 92. For example, when host sector addresses '4a+1' to '4b+2' are received from the host 90, they are converted to actual sector addresses 'a' and 'a+1' of the disk 92. And the sector data of this actual sector address is read to a buffer 93  
5 from the disk 92 (this is called staging).

In response to a read command, data of the specified host sector addresses '4a+1' to '4b+2' are obtained from the read sectors 'a' and 'a+1', and transferred to the host 90.

On the other hand, in response to a write command, this staged  
10 sector data 'a' and 'a+1' are updated by write data of the host sector addresses '4a+1' to '4b+2'. Next, the updated data are written to the disk 92 in actual sector units 'a' and 'a+1'.

In this way, the emulation operation reads the data of the address before and after that of the target address and updates it. When a write error occurs when writing to the disk 92, an  
15 alternate sector process is performed the same as in normal disk control. Generally, the objective of the alternate process is to save the host data without the assumption of a retry from the host. Therefore, conventionally, the write data from the host  
20 is written in an alternate area of the disk 92. For example, as shown in Fig. 11, when a write error occurs in the write data from host sector address '4a+2', the 2048-byte data of the sector address 'a' of this drive is saved in an alternate area.

However, in the prior alternate processing, when writing of  
25 alternation does not end normally, a retry operation called alternation is performed. However even in that case when all writing does not end normally, the writing process ends in error and

processing ends without all of the data being written.

Therefore, there is the problem that all of the data of that sector of the disk is lost from the drive and medium. Particularly in the emulation process, there is the problem that data of an address that is not the target for writing is deleted.

#### SUMMARY OF THE INVENTION

The objective of this invention is to provide a storage device and an emulation processing method for the storage device for protecting data even when a write error occurs during emulation.

Another objective of this invention is to provide a storage device and an emulation processing method for the storage device for saving all of the read data.

A further objective of this invention is to provide a storage device and an emulation processing method for the storage device for protecting data without a drop in emulation speed.

The emulation processing method for a storage device of this invention comprises: a step of reading data of an address that is specified with a second sector length unit from the storage medium with a first sector length unit; a step of rewriting the data that is read with the first sector length unit into data with the second sector length unit; a step of saving the data that is read or rewritten in the first sector length unit; a step of writing the rewritten data to the storage medium; and a step of registering the saved data as alternate data when the writing fails.

Moreover, the storage device of this invention comprises a

buffer memory, and a control circuit for reading the data of the address specified with a second sector length unit from the storage medium to the buffer memory with a first sector length unit, and then rewriting the data that have been read with the first sector length unit to data with the second sector length unit, and writing the rewritten data to the storage medium. In addition, the control circuit saves the data that is read or rewritten with the first sector length unit to save area, and registers that saved data as alternate data when writing fails.

In this form of the invention, first, before the data that are rewritten with the actual sector unit are written to the storage medium, the data that is read or rewritten with the first sector length unit are saved, so that when a write error occurs during writing, it is possible to save the data that is saved with the first sector length unit as alternate data. Therefore, by making this data of the first sector length unit as the alternate data, access with the first sector length unit is possible with this alternate data even when receiving a read command.

Second, the area that is used for rewriting is normally the work area of the memory, so normally it is cleared when writing to the disk in order for the next process. Therefore, it is not possible to use the data in the area used for rewriting as alternate data. In this invention, the data that are read or rewritten with the first sector length unit are saved, so the highly reliable alternation processing is possible.

In the emulation processing method of another form of the invention, the saving step comprises a step of saving the data

with the first sector length unit to the storage medium. Moreover, the control circuit saves the data with the first sector length unit to the storage medium.

In this form of the invention, a storage medium is used for saving the data. Therefore, when compared to using memory, data are securely saved and the medium can be used as is, as the alternate area.

The emulation processing method of another form of the invention comprises: a step of determining whether or not the corresponding data with the first sector length unit has all been rewritten from the target address with the second sector length unit for which there is a rewrite request; a step of executing the save step when all of the data have not been rewritten; a step of registering to the storage medium as alternate data , the data with the second sector length unit and for which there was a rewrite request when the writing fails, when all of the data have been rewritten.

Moreover, the control circuit determines whether or not the corresponding data with the first sector length unit has all been rewritten from the target address with the second sector length unit for which there is a rewrite request, execute the save step when all of the data have not been rewritten, and register to the storage medium as alternate data, the data with the second sector length unit and for which there was a rewrite request when the writing fails, when all of the data have be rewritten.

In this form of the invention, when all of the data with the first sector length are rewritten, performing of the save process

is stopped. In other words, when this save process is performed, the time required for the emulation process becomes longer. On the other hand, when all of the data with the first sector length are rewritten, the write data are data with the first sector length, so by using this fact, it is possible to prevent the emulation time from becoming longer due to performing the alternate processing.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of the file system of an embodiment of the invention.

Fig. 2 is configuration diagram of the optical disk drive in Fig. 1.

Fig. 3 is a drawing explaining the emulation process in Fig. 1.

Fig. 4 is a drawing explaining the sector of the disk in Fig. 2.

Fig. 5 is a flowchart of the initial process when inserting the medium of an embodiment of the invention.

Fig. 6 is a flowchart of the emulation process of an embodiment of the invention.

Fig. 7 is a drawing explaining the write emulation operation in Fig. 6.

Fig. 8 is drawing explaining the alternate area of the invention.

Fig. 9 is a flow chart of the emulation process of another



embodiment of the invention.

Fig. 10 is a drawing explaining the prior emulation process.

Fig. 11 is a drawing explaining the prior emulation operation.

5

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

The embodiments of this invention will be explained in the order of storage device, emulation process, and other emulation process.

### -- Storage Device --

Fig. 1 is a block diagram of the file system of an embodiment of the invention. Fig. 2 is a configuration drawing of the optical disk drive in Fig. 1. Fig. 3 is a drawing that explains the emulation process in Fig. 1, and Fig. 4 is a drawing explaining the sector format. Fig. 1 and Fig. 2 show a magneto-optical disk device as the storage device.

As shown in Fig. 1, a host 1 has an OS that is a UNIX OS and its recognizable sector length is 512 bytes. A magneto-optical disk device 10 is connected to the host 1. An interface 2 exchanges commands and data with the host 1. A MPU (microprocessor) 3 performs overall control of the magneto-optical disk device. A buffer memory 4 stores write data and read data as well as has a cache area 4-1.

A write LSI circuit 6 comprises a write modulation unit and laser diode control circuit. The write modulation unit modulates the write data, according to the type of magneto-optical disk,

into bit-position modulation (PPM) record (called mark record) data format or pulse-width modulation (PWM) record (called edge record) data format. The laser diode control circuit controls the intensity of the laser beam of the optical head of the drive 11 according to this modulated data.

A read LSI circuit 5 comprises: a read demodulation unit, having an AGC (auto gain control) circuit, filter and sector mark detection circuit; and a frequency synthesizer. The read demodulation unit generates a read clock and read data from an input ID signal or MO signal, and then demodulates the PPM data or PWM data into the original NRZ data.

The optical head 7 of the drive 11 detects the light returning from the magneto-optical disk 8, and inputs the ID signal/MO signal to a read circuit 5. As shown in Fig. 2, a spindle motor 40 rotates the magneto-optical disk 8. The spindle motor 40 is located in a housing 67. A magneto-optical disk cartridge 70 is inserted from an inlet 69. The magneto-optical disk 8 inside the cartridge 70 is rotated by the spindle motor 40.

The optical head 7 comprises a carriage 76 and fixed optical system 78. The carriage 76 moves along a rail 84 by a VCM (not shown in the figure) in a direction that crosses the tracks of the magneto-optical disk 8. The carriage 76 comprises an objective lens 80, direction conversion prism 82, focus actuator, track actuator, etc. The fixed optical system 78 comprises the aforementioned laser diode unit, ID/MO detector and 4-division detector.

As shown in Fig. 1, there is an emulation function in the

MPU 3. The basic operation of this emulation function is explained by Fig. 3 and Fig. 4.

As shown in Fig. 4, a sector of the magneto-optical disk 8 comprises a sector ID and user area (data area) with a sector length of 2048 bytes. There is a buffer area located outside of the user area. For comparison, the format of a 512-byte sector is shown. The 2048-byte sector has an advantage capacity wise.

On the other hand, the OS of the host 1 accesses files in 512-byte units (second sector length). The relationship between the sector recognized by the host and the actual sector of the disk is shown in Fig. 3, where the sector addresses of the host are '4a', '4a+1', '4a+2' and '4a+3' with respect to the actual sector of sector address 'a' of the disk.

First, the sector addresses from the host 1 are converted to an actual sector addresses of the disk 8. In Fig. 3, when the host sector addresses '4a+1' to '4b+2' of five sectors from the host 1 are received, they are converted to the two actual sector addresses 'a' and 'a+1' of the disk 8. Also, the sector data of the actual sector addresses are read from the disk 8 into a cache area 4-1 (this is called staging).

In response to a read command, data of the specified five host sector addresses '4a+1' to '4b+2' are extracted from the read sectors 'a' and 'a+1' and transferred to the host 1.

On the other hand, in response to a write command, the staged sector data 'a' and 'a+1' are rewritten by the write data of the five host sector addresses '4a+1' to '4b+2'. Next, the updated data or read data are written in an alternate area (for example

the buffer area of the disk 8 in Fig. 4) in units of actual sectors 'a' and 'a+1' (this is called saving).

Then, the updated data are written to the disk 8 in actual sector units 'a' and 'a+1'. When an error occurs while writing to the disk 8, the data in the alternate area are used as alternate data.

In this way, before writing to the disk 8, the updated data or read data are saved in the alternate area (for example the buffer area of the disk 8 in Fig. 4) in actual sector units 'a' and 'a+1', so when a write error occurs while writing, it is possible to save the saved data with first sector length units as alternate data, and even when receiving a read command, access in the first sector length units is possible using this alternate data.

Moreover, the area which is used for rewriting (cache area) is the work area, so normally it is cleared when writing to the disk 8 in order for the next process. Therefore, it is not possible to use data in the area used for rewriting as alternate data. In this invention, the data with the first sector length unit that are read or rewritten are saved, so highly reliable alternate processing is possible.

The storage device can use a storage device to store the data in sector units, that is, besides a magneto-optical disk, it is possible to use an optical disk, magnetic disk or the like. Moreover, an explanation is given of using a combination of a 2048-byte disk and 512-byte host, however other combinations are also possible. Furthermore, emulation processing is executed in the disk device, however, it can also be executed in another device

connected to the disk device, such as an SCSI board.

#### -- Emulation Processing --

Fig. 5 is a flowchart of the initial process after inserting  
5 the medium of an embodiment of the invention.

When the disk 8 is inserted, the control area of the disk  
8 is read and the medium identifier is checked. When the medium  
identifier indicates an emulation medium, it is recognized as an  
emulation medium and the emulation flag is turned ON. On the other  
10 hand, when the medium identifier indicates that the medium is not  
an emulation medium, then it is not recognized as an emulation  
medium and the emulation flag is turned OFF. Emulation processing  
is only performed when the emulation flag is ON.

Fig. 6 is a flowchart of the emulation process of an embodiment  
15 of the invention, and Fig. 7 is a drawing explaining the write  
emulation operation in Fig. 6.

(S1) The write command is received from the host 1. When  
it is a SCSI command, it is operation-code, logical block address  
(LBA) and number of blocks. In the write command, writing is  
20 specified in the operation-code. The logical block address and  
number of blocks are defined in 512-byte units as mentioned above.

(S2) When the MPU 3 receives the write command by way of the  
interface 2, it performs the size conversion process. In this  
case, the disk 8 has 2048-byte sectors so it is four times the  
25 host sector size. Therefore, the start sector address obtained  
from the LBA ('4a+1' in Fig. 3) is divided by four, and with the  
quotient remaining after discarding everything beyond the decimal

point, it is converted to the disk sector address ('a' in Fig. 3) to which the start host sector belongs. Also, the value beyond the decimal point is a fraction. Similarly, the end sector address ('4b+1' in Fig. 3) is obtained from the LBA and number of blocks. This is divided by four, and with the quotient remaining after discarding everything beyond the decimal point, (remainder), it is converted to the disk sector address ('a+1' in Fig. 3) to which the end host sector belongs. When there is a remainder (fraction) of this division, it indicates that part of the disk sector is not specified by the host sector.

(S3) The MPU 3 determines whether there is a fraction.

(S4) When there is a fraction, the MPU 3 stages the data of the converted disk sector addresses ('a' and 'a+1' in Fig. 3) from the disk 8 to the cache area 4-1 of the buffer memory 4. The write data sent from the host 1 are expanded in the receiving area (write data area) 4-2 (see Fig. 7) of the buffer 4. In addition the MPU 3 updates the sector data 'a', 'a+1' of the cache area 4-1 with the write data. In Fig. 3, there is a request to write the 5 sector/512 byte data from address [4a+1], so the data of addresses [4a+1] to [4b+1] are rewritten.

(S5) Next, the MPU 3 writes (saves) the sector data 'a', 'a+1' of the cache area 4-1 in the alternate area 8-1 of the disk 8. It is possible to use the sector buffer area in Fig. 4 as this alternate area. The MPU 3 determines whether saving was successful (no write errors). When saving was not successful, the host 1 is notified of the error and operation ends abnormally.

(S6) On the other hand, when saving is successful, the MPU

3 updates (rewrites) the sectors 'a', 'a+1' of the disk 8 with the sector data 'a', 'a+1' of this cache area 4-1. Whether an error occurred is determined by this rewriting.

(S7) When it is determined that a write error occurred, it updates the sectors 'a', 'a+1' of the disk 8 with the data saved in the alternate area (rewrite). The MPU 3 then determines whether this writing was successful (no write error), and when this writing is not successful, it activates the alternate area 8-1. In other words, the alternate address is set as the alternate information of the control area of the disk 8, and operation ends normally.

(S8) On the other hand, when writing or rewriting to the disk 8 in step 6 and step 7 is successful, there is no need for the data in the alternate area. Therefore, the alternate area 8-1 is deactivated. In other words, the flag of the alternate area is set to be invalid, and operation ends normally.

(S9) On the other hand, when the MPU 3 determines that there is no fraction in step S3, it stages the converted data of the disk sector address, from the disk 8 to the cache area 4-1 of the buffer memory 4. The write data that were sent from the host 1 are expanded in the receiving area (write data area) 4-2 (see Fig. 7) of the buffer 4. In addition, the MPU 3 updates the sector data in the cache area 4-1 with write data. In this case, since there is no fraction, in Fig. 3, the sector data 'a' is rewritten for the sector data of the four host sector addresses.

(S10) Next, the MPU 3 updates (rewrites) the sector of disk 8 with the sector data of this cache area 4-1. Whether a write error occurred is determined by this rewriting. When it is

determined that a write error occurred, normal alternate processing is performed. In other words, the MPU 3 writes the data in the write area 4-2 of the buffer 4 in the alternate area 8-1 of the disk 8. Also, it activates the flag of the alternate area and sets the address for the destination of the alternate information of the control area of the disk 8. Operation ends normally. When no write error occurs, operation ends normally without performing alternate processing.

Since data in the cache area in disk sector units are saved in the alternate area in this way, it is possible to obtain alternate data in disk sector units when updating the disk with data in the cache area in disk sector units even when a write error occurs.

The data in the cache area are cleared or written over as the aforementioned disk update operation is performed in order for the next process. Therefore, there is a possibility that the data in the cache area will change during writing even when a write error occurs. In this embodiment of the invention, the data are saved separate from the cache area, so it is possible to obtain alternate data in disk sector units.

Moreover, when a write error occurs, the saved data are rewritten to the disk. In other words, the data are immediately rewritten without performing alternate processing while updating the disk, even when a write error occurs. This is because there is a retry when updating the disk when a write error occurs, due to temporary power loss or the like. This makes it possible to prevent unnecessary use of the alternate area. In this case as well, the data in the cache area are cleared or written over as



the disk update operation is performed, in order for the next process. Therefore, the data of the alternate area become valid.

Furthermore, when there is no fraction, the saving process is not performed and normal alternate processing is performed. Since the saving process takes time, it is only performed when necessary. Therefore, it is possible to prevent the emulation time from becoming longer.

Moreover, the save area is used as the alternate area. Therefore, when the alternate is necessary, the write operation of the alternate area becomes unnecessary. This makes it possible to improve overall performance.

As a variation, with the alternate area as an area in the buffer 4, it is also possible to apply a method of writing to the alternate area of the disk when alternate processing is necessary. Furthermore, when using a SCSI board, the save area does not have to be a disk area.

Fig. 8 is a drawing explaining the alternate area of another embodiment of this invention. As shown in Fig. 8, the magneto-optical disk 8 is divided into a plurality of zones from the inside to the outside. In addition, as mentioned above, the writing frequency of each zone is changed in order to keep the writing density of each zone at an optimum value. There are buffer tracks on the boundary of these zones. A track is an empty area (number of blocks per sector x 4 logic tracks). Of these, the buffer track 1 is in the area where zones change, and there are parts that cannot be positioned, however, the buffer tracks 2, 3 are not limited and can be read from or written to.

This empty area is space due to the physical characteristics of the magneto-optical disk device, and it is assigned as the aforementioned alternate area.

Besides this, as mentioned above, since the alternate area is assigned in disk sector units, there is a possibility that there will not be enough alternate area. In that case, it is possible to assign the manufacture zone of the disk, or assign a test track or map area.

Also, in the aforementioned embodiment, the saving process is not performed when there is no fraction, however, when emphasizing the security of the data, it is possible to apply the saving process even when there is no fraction.

-- Another Emulation Process --

Fig. 9 is a flowchart of the emulation process of another embodiment of the invention.

(S11) A write command is received from the host 1.

(S12) When the MPU 3 receives a write command by way of the interface 2, it determines whether there is empty space in the storage area (disk or buffer). When there is no empty space in the storage area, the sector data cannot be saved, so it notifies the host 1 and advances to step S16.

(S13) When there is empty space in the storage area, the MPU 3 performs size conversion. It divides the starting address obtained from the LBA of the command by four, and converts it to a disk sector address to which the starting host sector belongs, and from the LBA and number of blocks, it converts it to the disk

sector address to which the end sector address belongs. It then stages the converted disk sector address data from the disk 8 to a cache area 4-1 in the buffer memory 4. Also, it writes this staged sector data beforehand to the storage area. It determines  
5 whether there is an error in writing to the storage area (write error). When there is no write error, it advances to step S17.

(S14) When there is a write error, it determines whether writing was retried a specified number of times. When it has not been retried the specified number of times it returns to writing in  
10 step S13.

(S15) When writing to the storage area is not successful, even after retrying the specified number of times, the MPU 3 notifies the host 1.

(S16) When the host 1 is notified that saving was not  
15 successful, the host 1 notifies whether it can or cannot perform emulation. In this case, when the host 1 cannot perform emulation, operation ends abnormally.

(S17) On the other hand, when the host 1 can perform emulation, it executes emulation. In other words, the write data sent from  
20 the host are expanded in the receiving area (write data area) of the buffer 4. In addition, the MPU 3 updates the sector data 'a', 'a+1' in the cache area 4-1 with the write data. In Fig. 3 since there is a 5 sectors/512 byte write request starting from address [4a+1], data of address [4a+1] to [4b +1] are rewritten.

(S18) Next, the MPU 3 updates (rewrites) the sectors 'a',  
25 'a+1' of the disk 8 with sector data 'a', 'a+1' in this cache area 4-1. The MPU 3 determines whether a write error occurred from

this rewriting. When no write error occurred, operation ends normally.

(S19) When the MPU 3 determines that a write error occurs, it then determines whether saving to the storage area was successful.

5 When saving to the storage area is successful, it notifies the host 1, then registers the storage area 8-1 in the alternate area. In other words, it sets the flag of the storage area to valid, and sets the alternate address for the alternate information in the control area of the disk 8. Operation then ends normally.

10 When saving to the storage area is not successful, the MPU 3 notifies the host 1 of that, and then the operation ends abnormally.

15 In this embodiment of the invention, the sectors that are read are stored beforehand and used as alternate data for emulation during writing. Since the state before writing can be saved when doing this, it is possible to prevent reading errors.

20 Moreover, since it performed before emulation, the emulation efficiency is good. Furthermore, when storage is not possible, the host is notified, so it is possible to control execution of emulation. When the read sectors cannot be stored when a writing error occurs, the host can perform processing such as a retry.

In addition to the embodiments described above, the invention can be changed as follows.

25 In the embodiments of the invention described above, an example of using a magneto-optical disk device as the storage device is explained, however, it is possible to use other storage devices such as a magnetic disk device, optical disk device, magnetic card

device, or optical card device.

The preferred embodiments of the present invention have been explained, however, the invention is not limited to these embodiments and can be embodied in various form within the scope  
5 of the present invention.

As explained above, the present invention has the following effects.

(1) Data defined a first sector length unit that are read or rewritten are saved before writing the rewritten data with actual  
10 sector units to the storage medium, so when a writing error occurs while writing, it is possible to save the saved data with first sector length units as alternate data. Therefore, since the data with first sector length units is taken to be alternate data, access in first sector-length units becomes possible using the alternate  
15 data even when a read command is received.

(2) Moreover, since the data with first sector length units that are read or rewritten are saved, highly reliable alternate processing is possible.

WHAT IS CLAIMED IS:

1. An emulation processing method for a storage device that accesses storage medium defined by a first sector length unit by a command from the host defined by a second sector length unit that is shorter than the first sector length, and comprising:

a step of reading data of an address that is specified with a second sector length unit from said storage medium defined by a first sector length unit;

a step of rewriting said read data defined by said first sector length unit into data defined by said second sector length unit;

a step of saving said data that is read or rewritten in the first sector length unit;

a step of writing said rewritten data to the storage medium; and

a step of registering said saved data as alternate data when said writing failed.

2. The emulation processing method for a storage device of claim 1, wherein said saving step comprises a step of saving the data with the first sector length unit to the storage medium.

3. The emulation processing method for a storage device of claim 1 further comprising:

a step of determining whether or not the corresponding all data with said first sector length unit is rewritten from a target address with said second sector length unit of write request;

a step of executing said save step when all of the data are

not rewritten; and

a step of registering to the storage medium as alternate data, said data with said second sector length unit of a write request when said writing failed, when all of the data is rewritten.

5

4. The emulation processing method for a storage device of claim 1, wherein said reading step comprises a step of staging said data with the first sector length unit to a cache area of a buffer memory from the storage medium.

5. The emulation processing method for a storage device of claim 1, wherein said registering step comprises a step of setting a valid flag to said save data.

6. The emulation processing method for a storage device of claim 1, further comprising;

a step of judging whether or not said saving step is success; and

a step of performing said rewriting step when said saving step is success.

7. The emulation processing method for a storage device of claim 1, wherein said reading step comprises a step of staging said data with the first sector length unit to a cache area of a buffer memory from the storage medium when receive a write command.

8. The emulation processing method for a storage device of

claim 1, further comprising;

a step of searching a save area when receive a write command from a host; and

a step of notifying that said saving step is not performed  
5 when said save area is not found by said searching step.

9. A storage device that accesses storage medium defined by a first sector length unit by a command from the host defined by a second sector length unit that is shorter than the first sector  
10 length, and comprising:

a buffer memory; and

a control circuit for reading data of the address specified with a second sector length unit from said storage medium to said  
15 buffer memory with a first sector length unit, and then rewriting said read data with the first sector length unit to data with the second sector length unit and writing said rewritten data to said storage medium; and wherein

said control circuit saves said data that is read or rewritten with the first sector length unit to a storage area, and registers  
20 said saved data as alternate data when said writing fails.

10. The storage device of claim 9, wherein said control circuit save the data with the first sector length unit to the storage medium.

25 11. The storage device of claim 9, wherein said control circuit further determine whether or not the corresponding all data with



said first sector length unit is rewritten from a target address with said second sector length unit of write request, execute said save step when all of the data are not rewritten, and register to the storage medium as alternate data, said data with said second  
5 sector length unit of a write request when said writing failed, when all of the data is rewritten.

12. The storage device of claim 9, wherein said control circuit stage said data with the first sector length unit to a cache area  
10 of said buffer memory from the storage medium.

13. The storage device of claim 9, wherein said control circuit set a valid flag to said save data for said registering.

14. The storage device of claim 9, wherein said control circuit judge whether or not said saving is success, and perform said  
15 rewriting when said saving is success.

15. The storage device of claim 9, wherein said control circuit stage said data with the first sector length unit to a cache area  
20 of said buffer memory from the storage medium when receive a write command.

16. The storage device of claim 9, wherein said control circuit search a save area when receive a write command from a host, and  
25 notify that said saving is not performed when said save area is not found by said searching.

17. The storage device of claim 9, further comprising a head  
for reading and writing said storage medium.

18. The storage device of claim 9, wherein said control circuit  
5 receive a write command from said host of UNIX OS.

## ABSTRACT

This invention relates to the emulation processing method of a storage device and the storage device for accessing storage medium from a host that has a different sector size, and where alternate processing is performed in the sector unit of storage medium even when there is a writing error when updating the disk. In a system where a second sector size of a host (1) is smaller than a first sector size of the storage medium (8), data that are staged in a buffer 4 according to a request from the host is rewritten using write data, and then staged sector data or the rewritten sector data are saved in an alternate area (8-1) and the storage medium (8) is updated with the rewritten sector data. Since the sector data before updating is saved beforehand, it is possible to perform alternate processing using the sector unit of the storage medium.

FIG.1

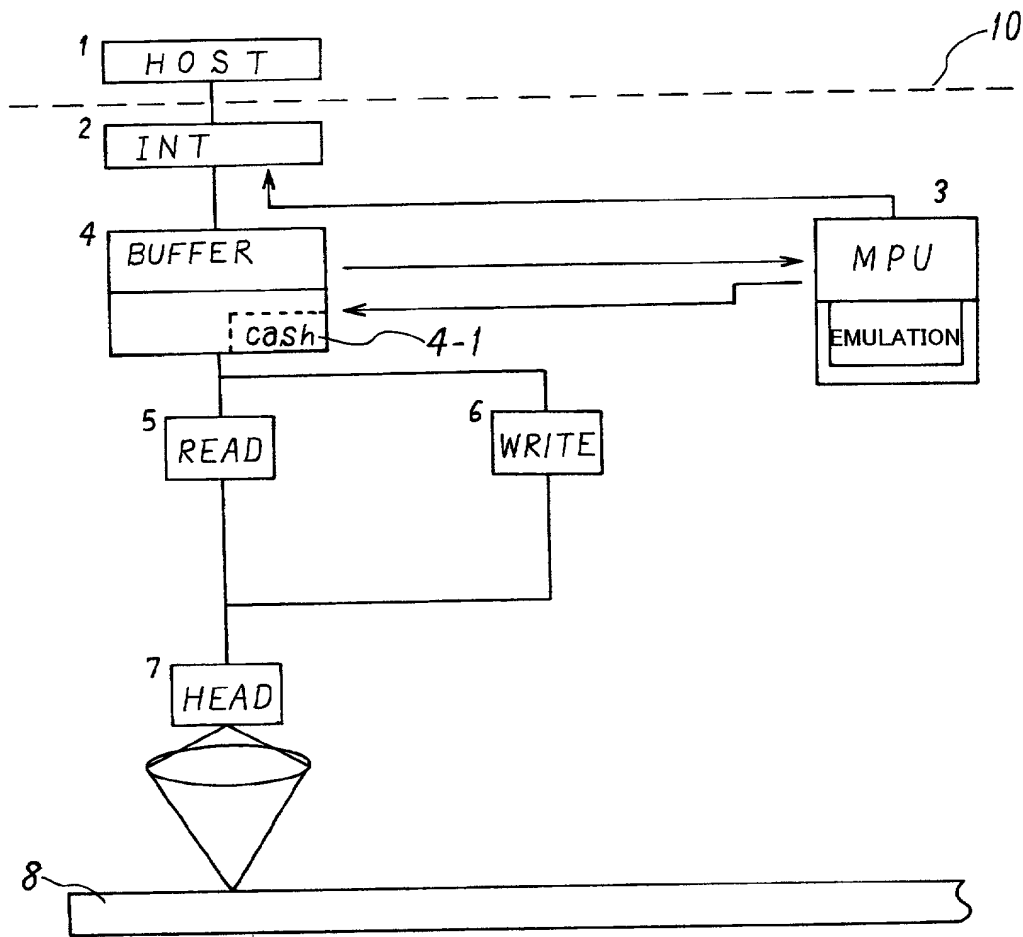


FIG. 2

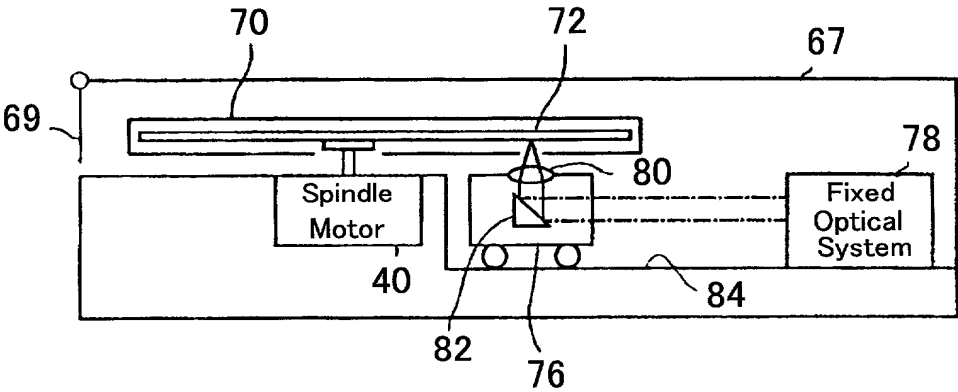


FIG.3

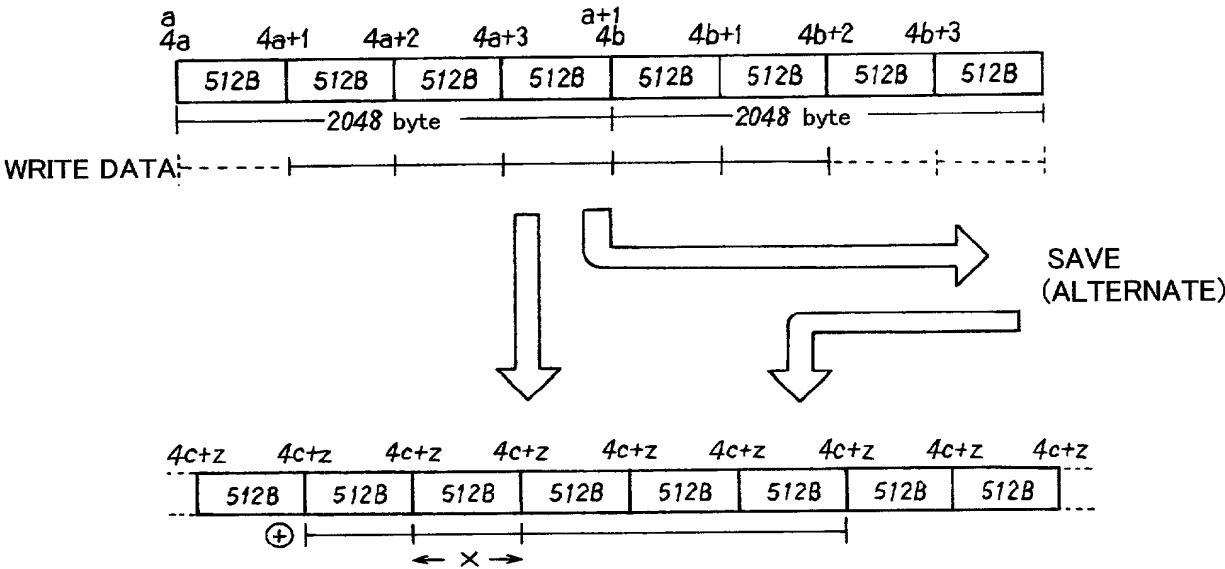


FIG.4

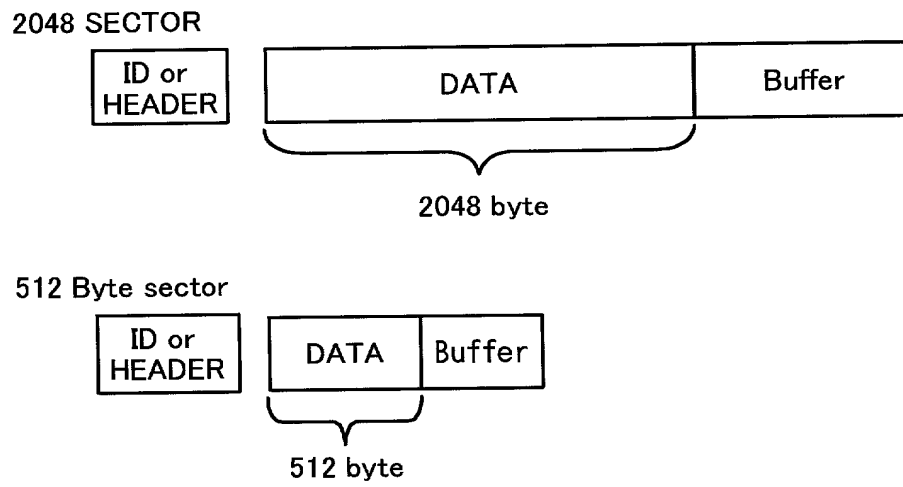


FIG.5

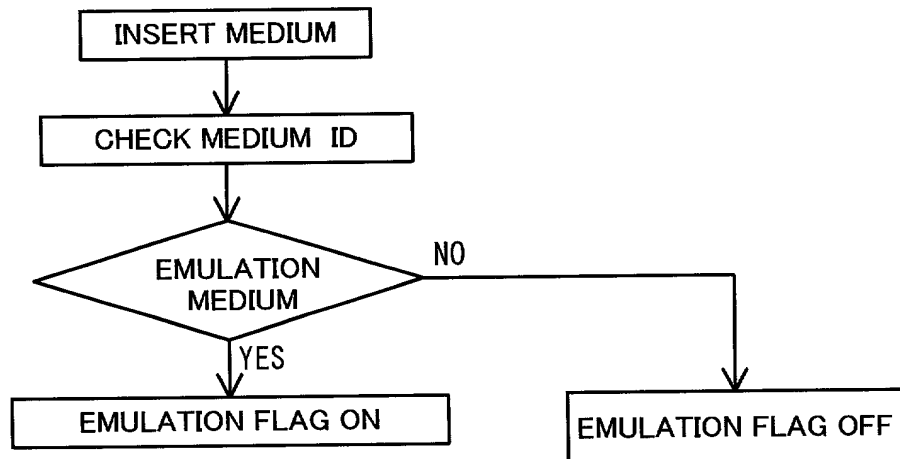


FIG. 6

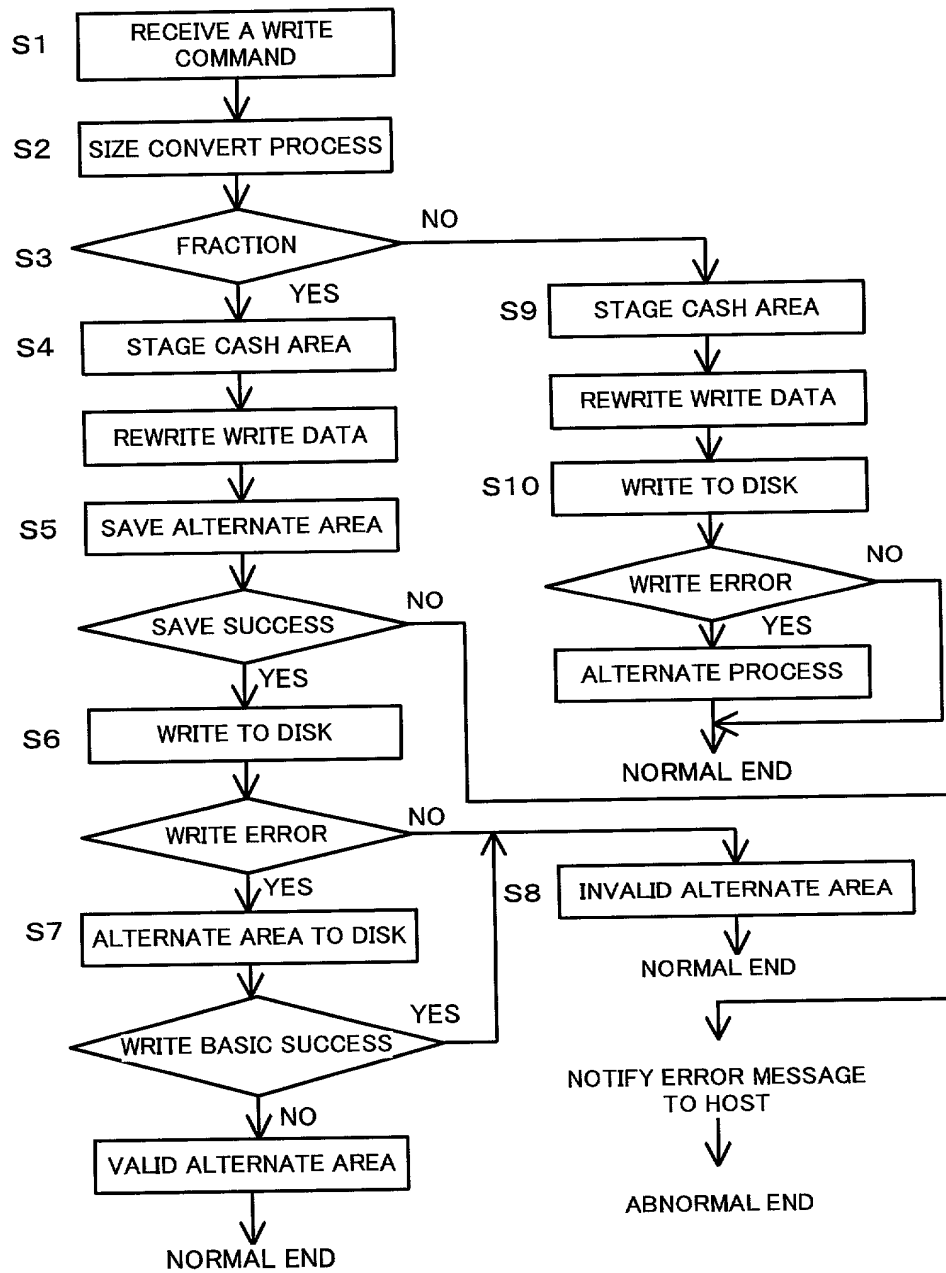


FIG.7

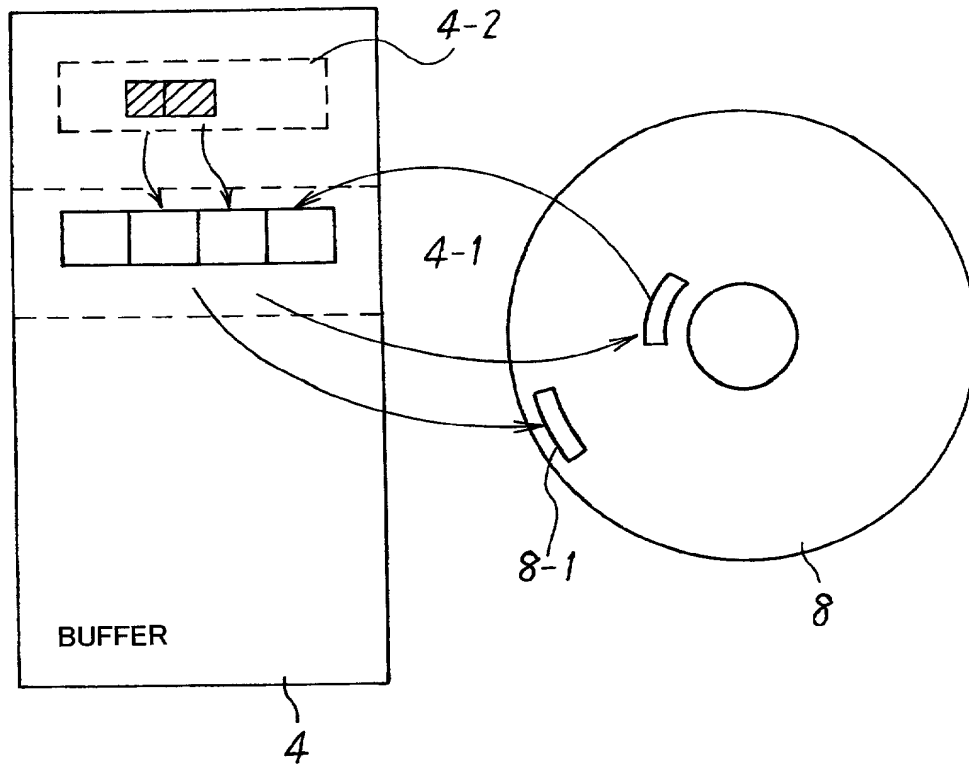




FIG.8

006250" 24492350

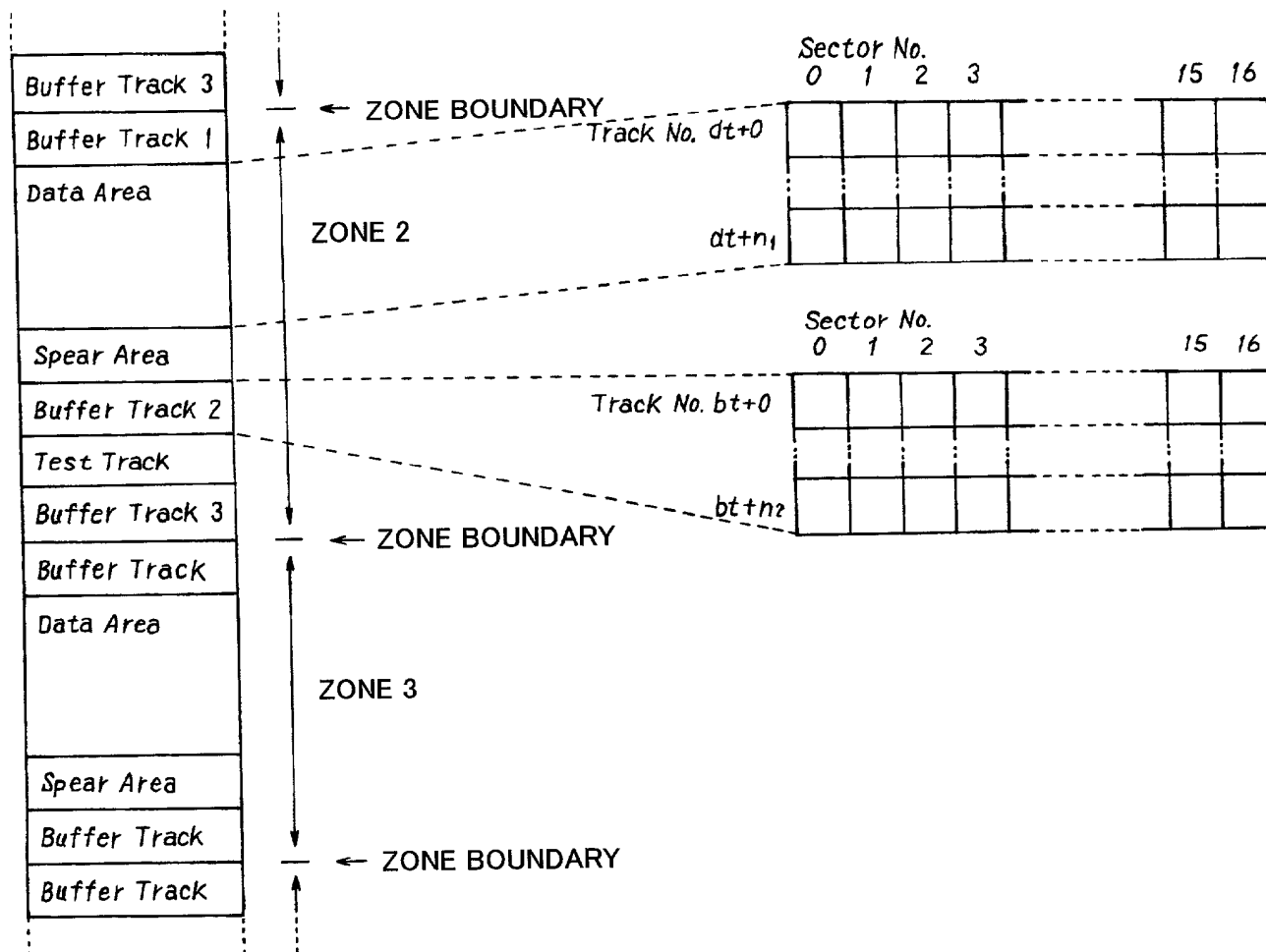
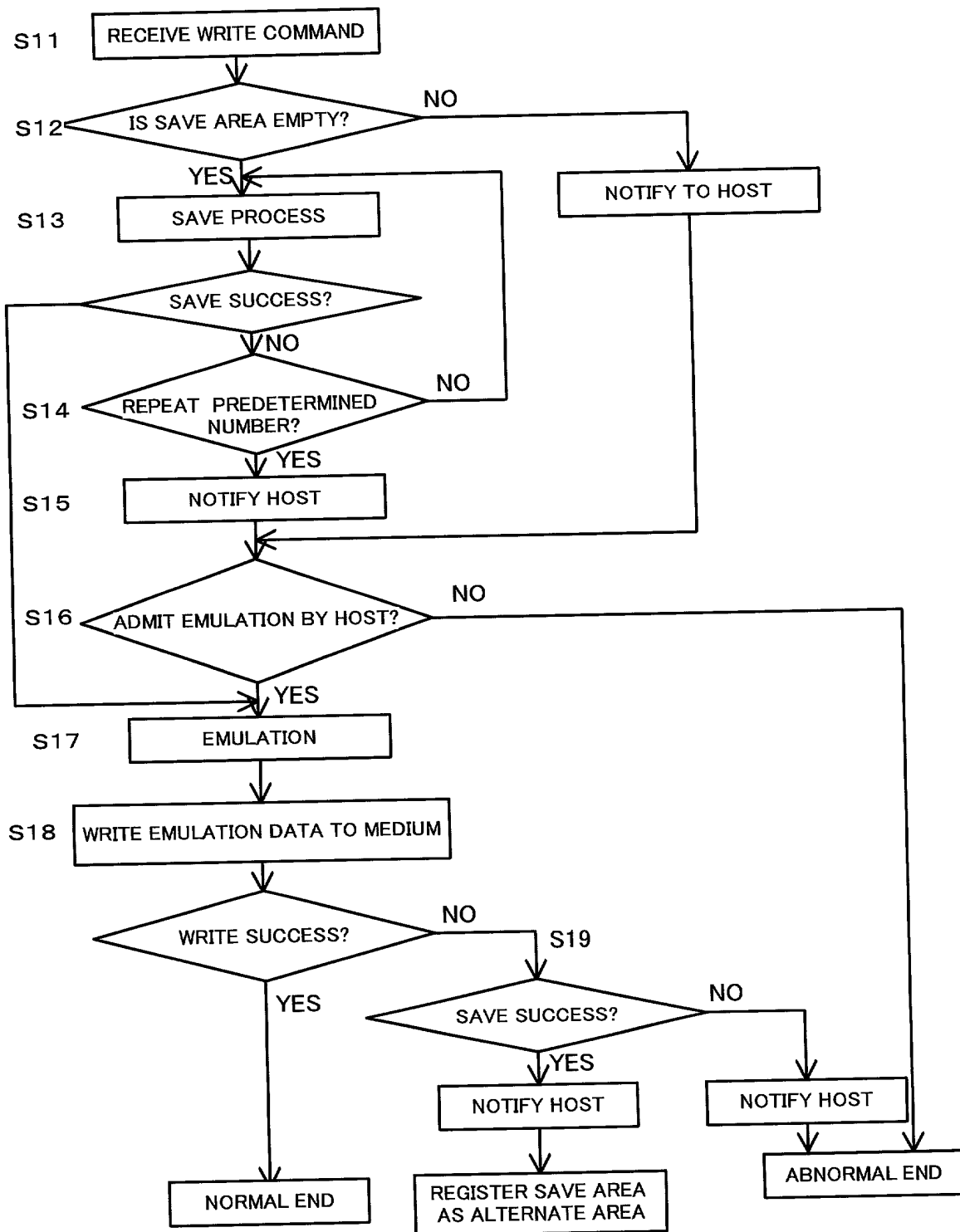
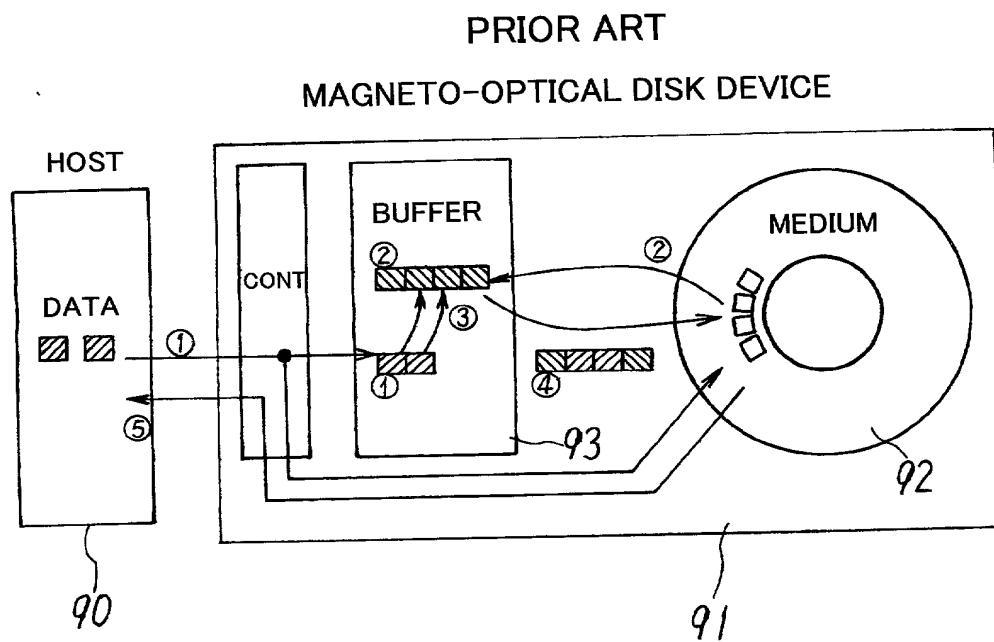


FIG.9

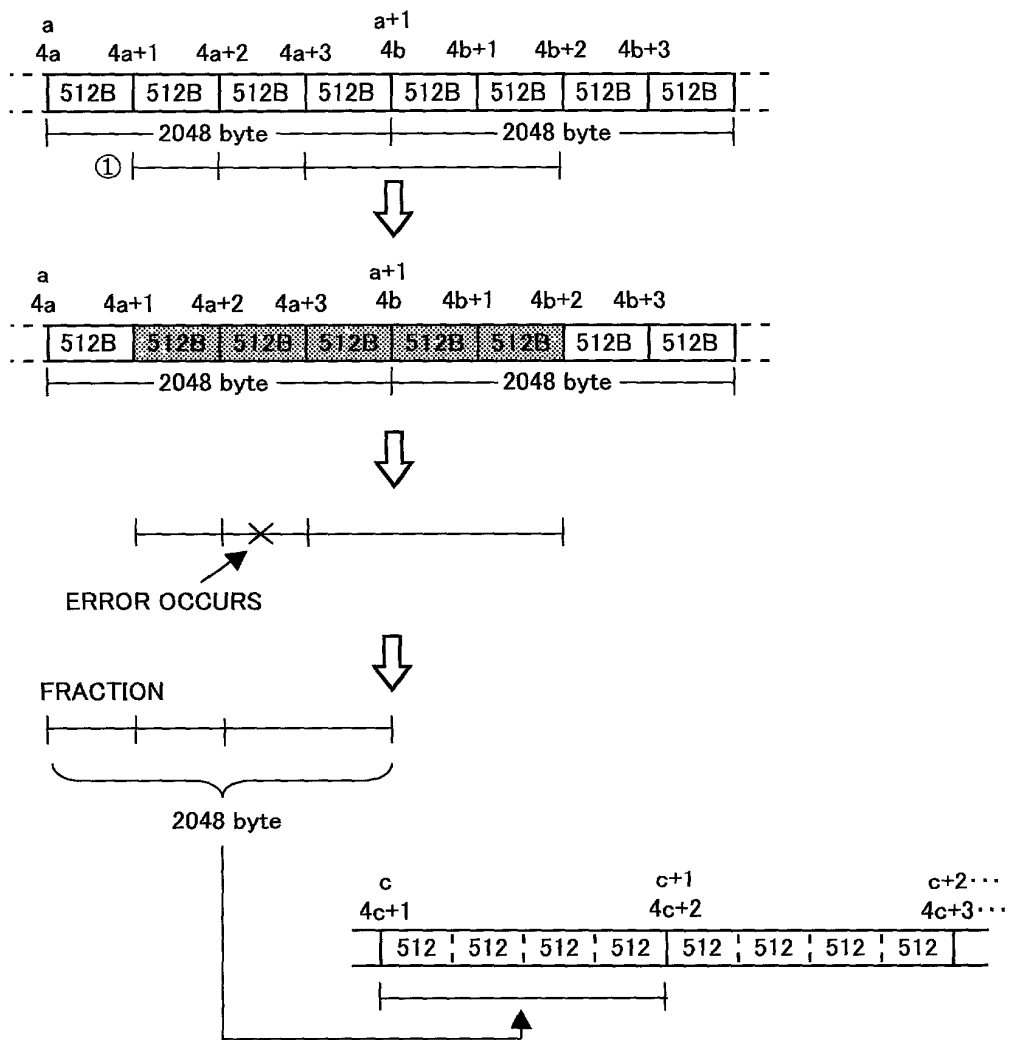


**THE**  
**NEW**  
**YORK**  
**PUBLIC**  
**LIBRARY**



# FIG. 11

PRIOR ART



## Declaration and Power of Attorney For Patent Application

### 特許出願宣言書及び委任状

### Japanese Language Declaration

### 日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

EMULATION PROCESSING METHOD FOR A STORAGE

DEVICE AND STORAGE DEVICE

上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

☐ \_\_月\_\_日に提出され、米国出願番号または特許協定条約国際出願番号を\_\_\_\_とし、  
(該当する場合) \_\_\_\_\_ に訂正されました。

☐ was filed on \_\_\_\_\_  
as United States Application Number or  
PCT International Application Number  
\_\_\_\_\_ and was amended on  
\_\_\_\_\_ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第37編第1条56項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

## Japanese Language Declaration (日本語宣言書)

私は、米国法典第35編119条(a)-(d)項又は365条(b)項に基づき下記の、米国外の国の少なくとも一カ国を指定している特許協力条約365(a)項に基づき国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している。本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

### Prior Foreign Application(s)

外国での先行出願  
11-360006

(Number)  
(番号)

Japan

(Country)  
(国名)

(Number)  
(番号)

(Country)  
(国名)

私は、第35編米国法典119条(e)項に基づいて下記の米国外特許出願規定に記載された権利をここに主張いたします。

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

私は、下記の米国法典第35編120条に基づいて下記の米国外特許出願に記載された権利、又は米国外を指定している特許協力条約365条(c)に基づき権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第35編112条第1項又は特許協力条約で規定された方法で先行する米国外特許出願に開示されていない限り、その先行米国外出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第37編1条56項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

私は、私自身の知識に基づいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じることに基づき表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行は米国法典第18編第1001条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed.

### Priority Not Claimed

優先権主張なし

17/12/1999

(Day/Month/Year Filed)  
(出願年月日)

☐

(Day/Month/Year Filed)  
(出願年月日)

☐

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.)  
(出願番号)

(Filing Date)  
(出願日)

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of application.

(Status: Patented, Pending, Abandoned)  
(現況: 特許許可済、係属中、放棄済)

(Status: Patented, Pending, Abandoned)  
(現況: 特許許可済、係属中、放棄済)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration  
(日本語宣言書)

委任状: 私は下記の発明者として、本出願に関する一切の  
手続を米特許商標局に対して遂行する弁護士または代理人  
として、下記の者を指名いたします。(弁護士、または代理  
人の氏名及び登録番号を明記のこと)

POWER OF ATTORNEY: As a named inventor, I hereby appoint  
the following attorney(s) and/or agent(s) to prosecute this  
application and transact all business in the Patent and Trademark  
Office connected therewith (list name and registration number)

Attorney	Reg. No.
Patrick G. Burns	29,367
Roger D. Greer	26,174
Lawrence J. Crain	31,497
Steven P. Fallon	35,132

Attorney	Reg. No.
James K. Folker	37,538
Jonathan D. Feuchtwang	41,017
B. Joe Kim	41,895
Joel H. Bootzin	42,343

直接電話連絡先: (名前及び電話番号)

Send Correspondence to:

Direct Telephone Calls to: (name and telephone number)

Patrick G. Burns, Esq.  
Greer, Burns & Crain, Ltd.  
Sears Tower - Suite 8660  
Chicago, IL 60606 (312) 993-0080

唯一または第一発明者名	Full name of sole or first inventor
発明者の署名	Katsumi SHIRAI
日付	Inventor's signature
住所	Katsumi Shirai
国籍	Residence
私書箱	Yokohama, Japan
	Citizenship
	Japan
	Post Office Address
	c/o FUJITSU PROGRAM LABORATORIES LIMITED,
	2-4-19, Shinyokohama Kouhoku-ku, Yokohama,
	222-0033 Japan
第二共同発明者	Full name of second joint inventor, if any
第二共同発明者	Kiyotaka FUKAWA
日付	Second inventor's signature
住所	Kiyotaka Fukawa
国籍	Residence
私書箱	Kawasaki, Japan
	Citizenship
	Japan
	Post Office Address
	c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome,
	Nakahara-ku, Kawasaki, 211-8588 Japan

(第三以降の共同発明者についても同様に記載し、署名を  
すること)

(Supply similar information and signature for third and subsequent  
joint inventors.)

第三共同発明者	Full name of third joint inventor, if any Takahiro NAKANO		
第三共同発明者	日付	Third inventor's signature Takahiro Nakano	Date 11/09/2000
住 所	Residence Hyogo, Japan		
国 籍	Citizenship Japan		
私書箱	Post Office Address c/o FUJITSU PERIPHERALS LIMITED, 35 Saho, Yashiro-cho, Kato-gun, Hyogo, 673-1447 Japan		
第四共同発明者	Full name of fourth joint inventor, if any		
第四共同発明者	日付	Fourth inventor's signature	Date
住 所	Residence		
国 籍	Citizenship		
私書箱	Post Office Address		

第五共同発明者	Full name of fifth joint inventor, if any		
第五共同発明者	日付	Fifth inventor's signature	Date
住 所	Residence		
国 籍	Citizenship		
私書箱	Post Office Address		
第六共同発明者	Full name of sixth joint inventor, if any		
第六共同発明者	日付	Sixth inventor's signature	Date
住 所	Residence		
国 籍	Citizenship		
私書箱	Post Office Address		

(第七以降の共同発明者についても同様に  
記載し、署名をすること)

(Supply similar information and signature for  
seventh and subsequent joint inventors.)